

**Soru 1:**

```
package finalSinavi;

public class Ogrenci {
    private String ad, soyad;
    private int numara, dogumYili;
    private double notOrtalamasi;

    public Ogrenci(String ad, String soyad, int numara) {
        this.ad = ad;
        this.soyad = soyad;
        this.numara = numara;
    }

    public double getNotOrtalamasi() {
        return notOrtalamasi;
    }

    public void setNotOrtalamasi(double notOrtalamasi) {
        this.notOrtalamasi = notOrtalamasi;
    }

    public void setDogumYili(int dogumYili) {
        this.dogumYili = dogumYili;
    }

    public String toString() {
        return numara + " " + ad + " " + soyad + " (" + dogumYili
            + ") " + notOrtalamasi;
    }

    /* Dosyaya kaydetmek için kalan alanlara erişim metotları.
       Veri gizleme ilkesi nedeniyle private olarak tanımlanmış alanlara
       BasitOgrenciOtomasyon nesnesi doğrudan erişemeyeceği için,
       Ogrenci nesnelerini diske kaydetmek için ek metotlar tanımlamak gerekir.
       Eğer metin dosyasına kayıt yapılacaksa dosyaSatiri, ikili dosyaya kayıt
       Yapılacaksa diğer 'getter' metotları kullanılmalıdır.
    */

    public String dosyaSatiri() {
        return numara + ";" + ad + ";" + soyad + ";" + dogumYili
            + ";" + notOrtalamasi;
    }

    public String getAd() {
        return ad;
    }

    public String getSoyad() {
        return soyad;
    }

    public int getNumara() {
        return numara;
    }

    public int getDogumYili() {
        return dogumYili;
    }
}
```

## Soru 2:

```
package finalSinavi;
import java.io.*;
import java.util.*;
public class BasitOgrenciOtomasyon {
    private Ogrenci[] ogrenciler;
    private int ogrenciDizini;
    private Scanner giris;

    public BasitOgrenciOtomasyon( int ogrenciSayisi ) {
        ogrenciler = new Ogrenci[ ogrenciSayisi ];
        ogrenciDizini = 0;
        giris = new Scanner( System.in );
    }
    public void ogrenciEkle( Ogrenci birOgrenci ) {
        ogrenciler[ ogrenciDizini ] = birOgrenci;
        ogrenciDizini++;
    }
    public void ogrencileriListele( ) {
        for( Ogrenci ogrenci: ogrenciler ) {
            if( ogrenci != null )
                System.out.println( ogrenci );
        }
    }
    public void ogrencileriKaydetMetin( String dosyaAdi ) {
        try {
            PrintWriter yazici = new PrintWriter(
                new FileWriter(dosyaAdi) );
            for( int i = 0; i < ogrenciDizini; i++ ) {
                yazici.println( ogrenciler[i].dosyaSatiri() );
            }
            yazici.close( );
        }
        catch (IOException e) {
            System.out.println( "Dosya işlemlerinde sorun." );
            e.printStackTrace();
        }
    }
    public void ogrencileriKaydetIkili( String dosyaAdi ) {
        char ayrac = ',';
        try {
            DataOutputStream yazici = new DataOutputStream(
                new FileOutputStream(dosyaAdi) );
            for( Ogrenci ogrenci: ogrenciler ) {
                yazici.writeInt( ogrenci.getNumara() );
                yazici.writeChars( ogrenci.getAd() );
                yazici.writeChar( ayrac );
                yazici.writeChars( ogrenci.getSoyad() );
                yazici.writeInt( ogrenci.getDogumYili() );
                yazici.writeDouble( ogrenci.getNotOrtalamasi() );
            }
            yazici.close();
        }
        catch (IOException e) {
            System.out.println( "Dosya işlemlerinde sorun." );
            e.printStackTrace();
        }
    }
}
```

### Soru 3:

```
//Buradaki kodların önemli bir kısmı main metoduna da çekilebilirdi.
//Ancak her işi main metodunda yapmaya alışmayın, çünkü nesneye yönelik
//programlarda işler nesneler üzerinden yürür. Main metodunun amacı ise
//programa ilk hareketi vermektir.

//Seçim yapma kodu main metoduna da çekilebilir.
public int secimYap( ) {
    int secenek = 0;
    while( secenek > 4 || secenek < 1 ) {
        System.out.println( "*** ANA MENÜ ***" );
        System.out.println( "1. Yeni öğrenci ekleme" );
        System.out.println( "2. Öğrencileri listeleme" );
        System.out.println( "3. Öğrencileri dosyaya kaydetme" );
        System.out.println( "4. Programdan çıkış" );
        System.out.print( "Seçiminizi Girin: " );
        //Scanner nesnesini burada da oluşturabilir.
        secenek = giris.nextInt( );
        if( secenek > 4 || secenek < 1 )
            System.out.println( "Lütfen 1-4 arası giriş yapınız." );
    }
    return secenek;
}

//Öğrenci bilgilerini alan kod main metoduna da çekilebilir.
//Ama o zaman main metodu static olduğundan BİR BasitOgrenciOtomasyon NESNESİ
//oluşturulup ogrenciEkle metodu onun üzerinden çalıştırılmalıdır.
public void ogrenciGirisi( ) {
    String ad, soyad;      int numara, yil;      double ortalama;
    System.out.print( "Öğrencinin numarasını giriniz:" );
    numara = giris.nextInt();
    System.out.print( "Öğrencinin adını giriniz:" );
    ad = giris.nextLine();
    System.out.print( "Öğrencinin soyadını giriniz:" );
    soyad = giris.nextLine();
    System.out.print( "Öğrencinin doğum yılını giriniz:" );
    yil = giris.nextInt();
    System.out.print( "Öğrencinin not ortalamasını giriniz:" );
    ortalama = giris.nextDouble();
    Ogrenci ogrenci = new Ogrenci( ad, soyad, numara );
    ogrenci.setDogumYili( yil );
    ogrenci.setNotOrtalamasi( ortalama );
    ogrenciEkle( ogrenci );
}

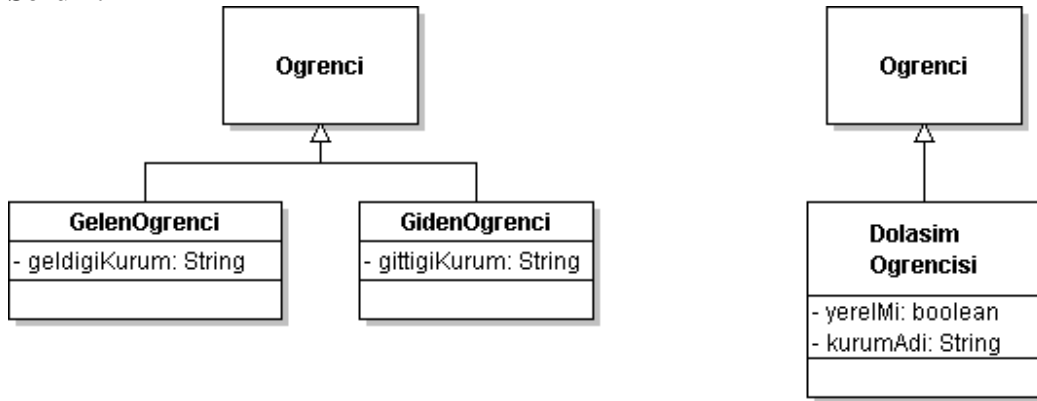
public void ogrenciKayit( ) {
    String dosyaAdi;
    System.out.print( "Dosya adını giriniz:" );
    dosyaAdi = giris.nextLine();
    ogrencileriKaydetMetin( dosyaAdi );
    System.out.println( "Öğrenciler dosyaya kaydedilmiştir." );
}
```

```

public static void main(String[] args) {
    BasitOgrenciOtomasyon anaProgram = new BasitOgrenciOtomasyon( 20 );
    //Bir BasitOgrenciOtomasyon nesnesi mutlaka oluşturulmalıdır.
    //Çünkü main static bir metottur ve üye metotlar doğrudan
    //çalıştırılmaz, mutlaka bir nesne üzerinden çağrılmalıdır.
    //Kaç öğrenci oluşturulacağı kullanıcıdan da alınabilirdi.
    int secenek = 0;
    while( secenek != 4 ) {
        //Dikkat: Eylemler bir BasitOgrenciOtomasyon örneği olan
        //anaProgram adlı nesne üzerinden yürütülüyor.
        secenek = anaProgram.secimYap( );
        switch( secenek ) {
            case 1:
                anaProgram.ogrenciGirisi( );
                break;
            case 2:
                anaProgram.ogrencileriListele( );
                break;
            case 3:
                anaProgram.ogrenciKayit( );
                break; } } } }

```

**Soru 4:**



İki çözümünden biri en uygun çözümdür. BasitOgrenciOtomasyon sınıfı özel bir tür öğrenci ile çalışacağı zaman tip dönüşümü yapmak zorundadır.